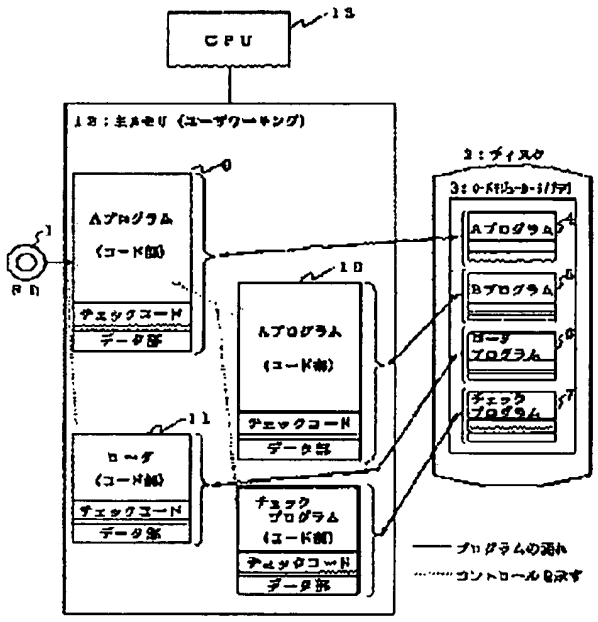
METHOD FOR MEMORY CHECK OF PROGRAM DURING SYSTEM OPERATION

Publication number: JP5028053
Publication date: 1993-02-05
Inventor: SATOU YOSHIHIRO
Applicant: OKI ELECTRIC IND CO LTD
Classification:
- International: ***G06F9/445; G06F11/10; G06F11/22; G06F11/30; G06F12/16;*** **G06F9/445; G06F11/10; G06F11/22; G06F11/30; G06F12/16;** (IPC1-7): G06F9/445; G06F11/10; G06F11/22; G06F11/30; G06F12/16
- European:
Application number: JP19910207629 19910724
Priority number(s): JP19910207629 19910724

Abstract of **JP5028053**

PURPOSE:To detect the abnormality in the early stage to prevent unnecessary malfunction and to easily analyze the cause of abnormality by calculating a check code to preserve the correct code at the time of first, read of a program to a library and calculating the code again to compare it, at the time of setting this program to the state waiting for execution. CONSTITUTION:In the system which does not have a hard memory check function or a memory protect function during the system operation, a program as the check object is read into a load module library 3 from a floppy disk 1 through a main memory 12 by a loader. At this time, the loader calculates the total sum of the code part of the program and preserves it as the check code in the trailing part of the code part. An execution object program is executed as a task; and in the wait state, the check code is calculated and is compared with the preserved code, and the abnormality is judged and the system is stopped to preserve abnormality information of the address, the code, or the like which may be destroyed if they do not coincide with each other.



Data supplied from the *esp@cenet* database - Worldwide

(19)日本国特許庁（ＪＰ） (12) 公開特許公報（A） (11)特許出願公開番号

特開平5－28053

(43)公開日 平成５年(1993)２月５日

(51)Int.Cl.[5]		識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ	12/16	３１０ Ｂ	7629－5B		
	9/445				
	11/10	３１０ Ｂ	7832－5B		
	11/22	３６０ Ｌ	9072－5B		
			8944－5B	Ｇ０６Ｆ 9／06 ４２０ Ｓ	

審査請求 未請求 請求項の数１(全 ６ 頁) 最終頁に続く

(21)出願番号 特願平3－207629

(22)出願日 平成３年(1991)７月24日

(71)出願人 000000295
沖電気工業株式会社
東京都港区虎ノ門１丁目７番12号

(72)発明者 佐藤 賀浩
東京都港区虎ノ門１丁目７番12号 沖電気工業株式会社内

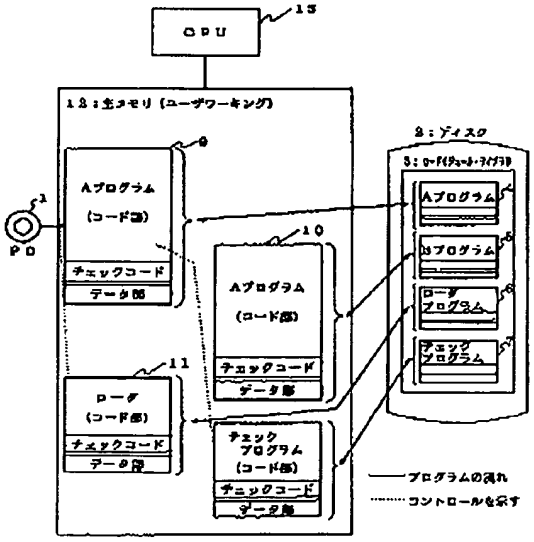
(74)代理人 弁理士 熊谷 隆 (外１名)

(54)【発明の名称】 システム運転中のプログラムのメモリチエツク方法

(57)【要約】

【目的】 ハード的なメモリチェック機能やメモリプロテクト機能がないシステムで、メモリの破壊を早期に発見し、無用な誤動作を防ぎ、破壊情報を保存し異常原因の解析を容易にすることを目的とする。

【構成】 本発明によるメモリチェック方法は、ロードモジュール・プログラムが格納されているＦＤ１と、ライブラリー等があるディスク２と、ロードモジュール・ライブラリー３と、チェック対象のＡプログラム４，９と、Ｂプログラム５，１０等と、チェックプログラム７，８と、ローダプログラム６，１１と、主メモリ１２と、ＣＰＵ１３で構成される。



本発明によるメモリチェック時のシステムの状態を示す図

1

【特許請求の範囲】

【請求項1】 システム運転中に、ハード的なメモリチェック機能やメモリプロテクト機能がない計算機システムのメモリチェックの方法において、

ロードモジュール・プログラムをライブラリーに読み込むとき、ロードモジュール・プログラムのコード部のチェックコードを計算し保存しておき、システム運転中に、ＣＰＵがタスク処理をする空き時間を利用して、対象とするタスクのロードモジュール・プログラムのコード部のチェックコードを計算し、その結果を前記保存されているチェックコードと比較し、もし値が異なればロードモジュール・プログラムが破壊されたと判断して、システムを停止させ、不必要な誤動作を防ぎ、異常情報を保存し障害解析を容易にさせることを特徴とするメモリチェック方法。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、ソフトウェアの動作中におけるロードモジュール・プログラム（以下プログラムと書く）のコード部の破壊を早期に検出するプログラムのメモリチェック方法に関するものである。なお、ここで適用されるシステムは、システム運転中に、ハード的なメモリチェック機能、メモリプロテクト機能がないシステムで、対象となるプログラムはコード部とデータ部が分離されているものとする。

【０００２】

【従来技術】従来ハード的なメモリプロテクト機能を持たないシステムでは、動作中にプログラムのコード部が破壊された場合、その時点では検出されずシステムダウン後、主メモリやディスクの情報を解析して検出されるものであった。

【０００３】

【発明が解決しようとする課題】しかしながら従来技術では、プログラムのコード部が破壊された時点では異常信号が検出されず誤動作の検出が遅れ、システムの運用に支障をきたす。又コード部が破壊されたままのプログラムが実行され、更に他のメモリーを破壊する場合がある。この場合は、システムダウン後に主メモリやディスクの情報を解析しても最初の異常原因の解明が不可能になる、という問題点があった。

【０００４】本発明は上述の点に鑑みてなされたもので、

①上記問題点の影響を最小限に留めるように異常を早期検出し、②破壊された後の無用の誤動作を防ぎ、③システムダウン後の異常原因の解析を容易にする、為のシステム運転中のプログラムのメモリチェック方法を提供することを目的とする。

【０００５】

【課題を解決するための手段】上記課題を解決するため本発明は、図１に示すチェックプログラムを設けた。まず

2

チェック対象となるＡプログラムはフロッピーディスクＦＤ１からローダにより主メモリを通してロードモジュール・ライブラリー３に読み込まれる。この時にローダは、プログラムのコード部の総和を計算しチェックコードとしてコード部の後部に保存する。図２にその動作をフローチャートに示す。その他のプログラムも同様にロードモジュール・ライブラリー３に読み込まれる。ロードモジュール・ライブラリーは主メモリ１２にあってもよい。

【０００６】プログラムがタスクとして実行されるとき、チェック対象となるプログラムは主メモリ１２に呼び出されて実行される。実行対象プログラムが待ち状態になった時、ＣＰＵが他に実行するタスクがないとき、チェックプログラムを主メモリに呼び出しタスクとして実行する。図３にその動作のフローチャートを示す。この時チェック対象となるプログラムのチェックコードを計算し前記保存されているチェックコードと比較し、もしその値が等しくなければ、そのチェック対象プログラムは破壊されたと判断し、システムを停止し破壊されたアドレスとかコード等の異常情報を保存する。以上の作業をすべてのチェック対象のプログラムに対して行う。チェックプログラム自身もチェック対象となる。

【０００７】

【作用】本発明によれば、チェック対象プログラムは、最初にライブラリーに読み込まれる時にチェックコードが計算され正しいチェックコードとして保存される。このプログラムがタスクとして実行中の待ち状態にされた時、再度チェックコードが計算されて両チェックコードが比較されるので、この期間にプログラムが破壊されたか否かが分かる。

【０００８】このチェックプログラムは、優先順位が最も低いタスクとして実行されるのでシステムの性能には悪影響を与えない。破壊される原因としてハード的な故障やソフト的には自分自身のタスクの暴走や他のタスクの暴走で壊される場合がある。例えば、図１のＡプログラム９が自分自身を壊す場合もあり、Ｂプログラム１０により壊される場合もあるが、何れの原因で破壊されてもチェックコードで検出可能である。

【０００９】

【実施例】以下本発明の一実施例を図面に基づいて詳細に説明する。図１は本発明によるチェック時のシステムの状態を示す図、図２はチェック対象プログラムをライブラリに読み込むときのチェックコードを計算し付加するフローチャート、図３はチェックプログラムのフローチャートである。

【００１０】プログラムはフロッピーディスクＦＤ１から主メモリ１２を通してディスク２のロードモジュール・ライブラリー３に読み込まれる。フロッピーディスクＦＤ１は、ディスク２上であっても他の外部メモリであってもよく、ロードモジュール・ライブラリ３は主メモ

リ12にあってもよい。ロードモジュール・ライブラリー3にはチェック対象となるAプログラム4、Bプログラム5、ローダプログラム6、チェックプログラム7等の多くのプログラムが格納されている。

【0011】これらのプログラムは、コード部とデータ部が分離されコード部の後部にチェックコードが格納される構造になっている。これらのプログラムが実行されるときは、主メモリ12上に必要なエリヤ8，9，10，11が確保され、ディスク2のロードモジュール・ライブラリー3からローダプログラム11によって主メモリ12上に呼び出され、必要な環境条件が準備されてCPU13によってタスクとして実行される。各タスクは、優先順位の高い順に実行される。チェックプログラムを実行するタスクは、優先順位を最低に設定しておく。こうすることにより他のタスクの実行を邪魔することなくチェックすることが可能となる。

【0012】本発明によるメモリチェックは、上記に詳述した状態で、チェックプログラム7で行われ、最初のチェックコードの付加は、ローダプログラム6によって行われる。図2に基づいてチェックコード付加の動作を説明する。プログラムをロードモジュール・ライブラリーに読み込む際、コード部の先頭アドレスを取り出す（ステップST1）。次にコード部の最終アドレスを取り出す（ステップST2），コード部の先頭から最終までを加算し（ステップST3）チェックコードとして所定の場所に格納する（ステップST4）。

【0013】次にチェックプログラムの動作を図3に基づいて説明する。プログラムインデックスを初期化する（ステップST11）。同インデックスで指定されたチェック対象のプログラムのコード部の先頭アドレスと最終アドレスを取り出す（ステップST12）。図2と同じ方法でチェックコードを計算し（ステップST13）、保存されているチェックコードと比較する（ステップST14）。もし同じであれば、今のプログラムが最終プログラムかどうかチェックし（ステップST15）、もし最終プログラムでなければインデックスを次

に進め（ステップ16）、最終プログラムであればステップST1より繰り返す。ステップST4でチェックコードが等しくなければ異常と見做し、破壊されたアドレスやコードなどの異常情報を保存してシステムを停止する（ステップST17）。

【0014】

【発明の効果】以上、詳細に説明したように本発明によれば、下記のような効果が期待できる。

①プログラムのコード部の破壊を早期に検出できる。

②破壊されたプログラムが実行されることによる誤動作を防ぐことができる。

③破壊されたアドレスとかコード等の破壊情報が保存され、無用な誤動作を最小限に抑えるためにシステム停止後の原因解析が容易になる。

【図面の簡単な説明】

【図1】本発明によるメモリチェック時のシステムの状態を示す図。

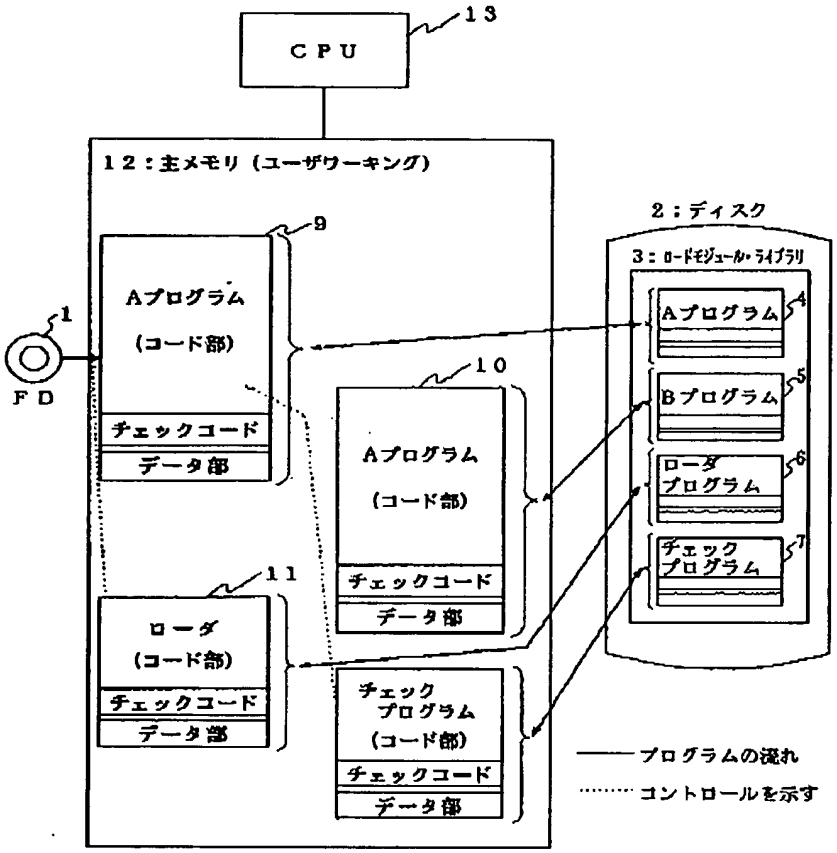
【図2】本発明に使用するチェックコード計算のフローチャート。

【図3】本発明によるメモリチェックプログラムのフローチャート。

【符号の説明】

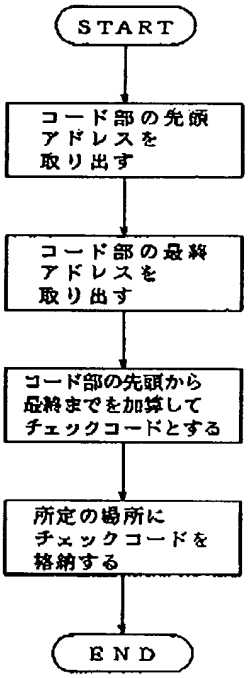
1	フロッピーディスク
2	ディスク
3	ロードモジュール・ライブラリー
4	Aプログラム
5	Bプログラム
6	ローダプログラム
7	チェックプログラム
8	チェックプログラム
9	Aプログラム
10	Bプログラム
11	ローダ
12	主メモリ
13	CPU

【図1】



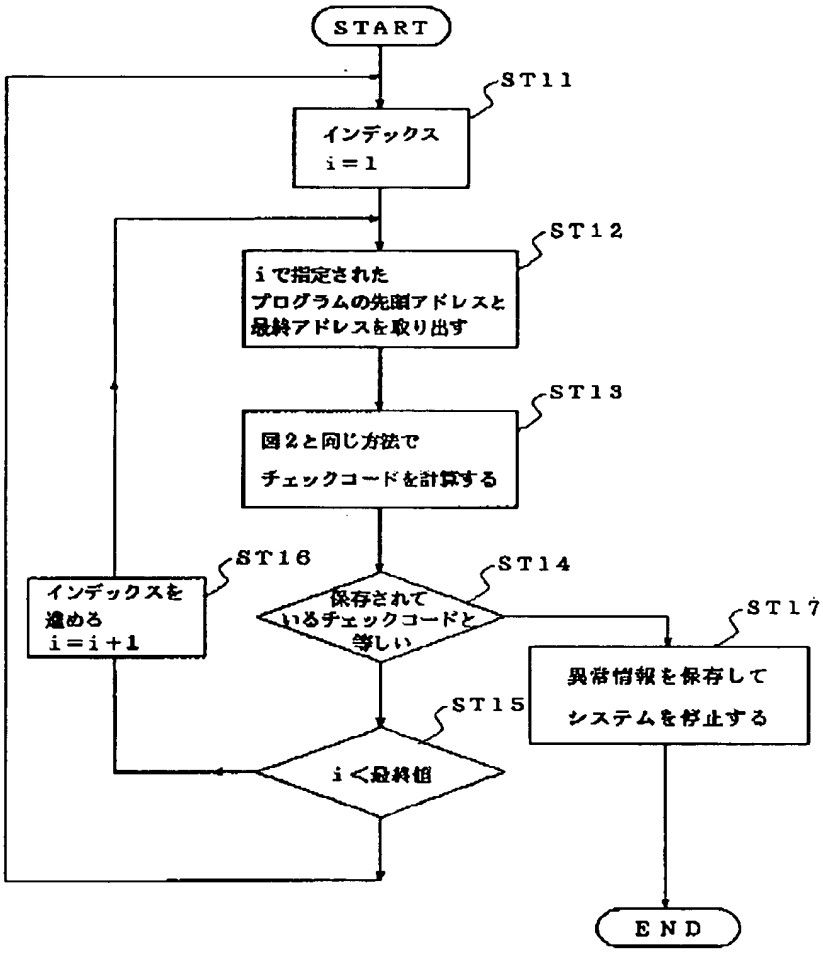
本発明によるメモリチェック時のシステムの状態を示す図

【図2】



チェックコード計算のフローチャート

【図3】



フロントページの続き

(51) Int. Cl.[5]	識別記号	庁内整理番号	F I	技術表示箇所
G06F 11/30	320 C	8725－5B		